NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE

(Accredited by NAAC,ISO 9001-2015 certified, Approved by AICTE New Delhi, Affiliated to KTU)

Pampady, Thiruvilwamala(PO), Thrissur(DT), Kerala 680 588

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

CSL 203 OBJECT ORIENTED PROGRAMMING LAB(JAVA)

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- Established in: 2002
- Courses offered : B.Tech in Computer Science and Engineering

  M.Tech in Computer Science and Engineering

  M.Tech in Cyber Security
- Approved by AICTE New Delhi and Accredited by NAAC
- Certified by ISO 9001-2015.
- Affiliated to the A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Team work and leadership qualities.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering

problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write

effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‟s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

List of Experiments

1. Write a Java Program to find the frequency of a given character in a string.
2. Write a Java program to multiply two given matrices.
3. Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'print-Salary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. (Exercise to understand inheritance).
4. Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures. (Exercise to understand polymorphism).
5. Write a Java program that read from a file and write to file by handling all file related exceptions.
6. Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util).
7. Write a Java program that shows thread synchronization.
8. Write a Java program that shows the usage of try, catch, throws and finally.
9. Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing.
10. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.
11. Write a Java program for the following:
    1) Create a doubly linked list of elements.
    2) Delete a given element from the above list.
    3) Display the contents of the list after deletion.
12. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.

CSL 203	OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	0	0	3	2	2019

Preamble: The aim of the course is to provide hands-on experience to the learners on various object oriented concepts in Java Programming. This course helps the learners to enhance the capability to design and implement various Java applications for real world problems.

Prerequisite: Topics covered under the course Programming in C (EST 102)

Course Outcomes:

At the end of the course, the student should be able to

CO1	Implement the Object Oriented concepts - constructors, inheritance, method overloading & overriding and polymorphism in Java (Cognitive Knowledge Level: **Apply**)
CO2	Implement programs in Java which use datatypes, operators, control statements, built in packages & interfaces, Input/Output streams and Files (Cognitive Knowledge Level: **Apply**)
CO3	Implement robust application programs in Java using exception handling (Cognitive Knowledge Level: **Apply**)
CO4	Implement application programs in Java using multithreading and database connectivity (Cognitive Knowledge Level: **Apply**)
CO5	Implement Graphical User Interface based application programs by utilizing event handling features and Swing in Java (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	☑	☑	☑	☑	☑			☑		☑		☑
CO2	☑	☑	☑	☑	☑			☑		☑		☑
CO3	☑	☑	☑	☑	☑			☑		☑		☑
CO4	☑	☑	☑	☑	☑			☑		☑		☑
CO5	☑	☑	☑	☑	☑			☑		☑		☑

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Test - Internal Exam (Percentage)	End Semester Examination (Percentage)
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab **:** Linux

Compiler/Software to Use in Lab **:** gcc, javac, jdk, jre, Eclipse, NetBeans, MySQL / PostgreSQL.

Programming Language to Use in Lab **:** Java

Fair Lab Record:

All Students attending the Object Oriented Programming Lab (in Java) should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Operations Performed, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

SYLLABUS

The syllabus contains six sessions (A, B, C, D, E, F). Each session consists of three concrete Java exercises, out of which at least two questions are mandatory.

(A) Basic programs using datatypes, operators, and control statements in Java.

1) Write a Java program that checks whether a given string is a palindrome or not.
   Ex: MALAYALAM is palindrome.
2) Write a Java Program to find the frequency of a given character in a string. **
3) Write a Java program to multiply two given matrices. **

(B) Object Oriented Programming Concepts: Problem on the use of constructors, inheritance, method overloading & overriding, polymorphism and garbage collection:

4) Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. (Exercise to understand inheritance). **
5) Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures. (Exercise to understand polymorphism). **
6) Write a Java program to demonstrate the use of garbage collector.

(C) Handling different types of files as well as input and output management methods:

7) Write a file handling program in Java with reader/writer.
8) Write a Java program that read from a file and write to file by handling all file related exceptions. **
9) Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util). **

(D) Exception handling and multi-threading applications:

10) Write a Java program that shows the usage of try, catch, throws and finally. **
11) Write a Java program that implements a multi-threaded program which has three threads. First thread generates a random integer every 1 second. If the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.
12) Write a Java program that shows thread synchronization. **

(E) Graphics Programming:

13) Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing. **
14) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts. **
15) Write a Java program to display all records from a table using Java Database Connectivity (JDBC).

(F) Standard Searching and Sorting Algorithms using data structures and algorithms learned from course Data Structures (**CST 201**):

16) Write a Java program for the following: **
    1) Create a doubly linked list of elements.
    2) Delete a given element from the above list.
    3) Display the contents of the list after deletion.
17) Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order. **
18) Write a Java program that implements the binary search algorithm.

** Mandatory

PRACTICE QUESTIONS

1) Write a Java program to reverse an given string.
2) Write a Java program to display the transpose of a given matrix.
3) Write a Java program to find the second smallest element in an array.
4) Write a Java program to check whether a given number is prime or not.
5) Write a Java program to calculate the area of different shapes namely circle, rectangle, and triangle using the concept of method overloading.
6) Write two Java classes Employee and Engineer. Engineer should inherit from Employee class. Employee class to have two methods display() and calcSalary(). Write a program to display the engineer salary and to display from Employee class using a single object instantiation (i.e., only one object creation is allowed).
   - display() only prints the name of the class and does not return any value. Ex. " Name of class is Employee."
   - calcSalary() in Employee displays "Salary of employee is 10000" and calcSalary() in Engineer displays "Salary of employee is 20000."
7) Write a Java program to illustrate Interface inheritance.
8) Write a Java program that shows how to create a user-defined exception.
9) Write a Java program to create two threads: One for displaying all odd number between 1 and 100 and second thread for displaying all even numbers between 1 and 100.
10) Write a Java program that shows thread priorities.
11) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
12) Write a Java program that displays the number of characters, lines and words in a text file.
13) Write a Java program for handling mouse events.
14) Write a Java program for handling key events using Adapter classes (general).
15) Write a Java program that allows the user to draw lines, rectangles and ovals.
16) Write a Java Swing program to print a wave form on the output screen.
17) Write a program to accept rollno, name, CGPA of "n" students and store the data to a database using JDBC connectivity. Display the list of students having CGPA greater than 7. (Use MySQL /PostgreSQL).
18) Write a Java program to implement Heap sort algorithm using array.

Instructions

1. You should come prepared for you lab session to properly utilize the maximum time of Lab session.
2. You should attempt all lab exercises/assignments given in the list (session wise).
3. You may seek assistance in doing the lab exercises from the available lab instructor.
4. There should be proper comments for description of the problem, requirement of class, function etc. Proper comments are to be provided as and when necessary in the programming. This will develop a good practice in you for rest of your professional life.
5. Your program should be interactive and properly documented with real Input/ Output data.
6. Proper management of file of Lab record is necessary. Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given.

Exp No. Date:

Frequency of a Given Character in a String

Aim: Write a Java Program to find the frequency of a given character in a string.

Algorithm:

1. Start
2. Define a string.
3. Define an array freq with the same size of the string.
4. Two loops will be used to count the frequency of each character. Outer loop will be used to select a character and initialize element at corresponding index in array freq with 1.
5. Inner loop will compare the selected character with rest of the characters present in the string.
6. If a match found, increment element in freq by 1 and set the duplicates of selected character by '0' to mark them as visited.
7. Finally, display the character and their corresponding frequencies by iterating through the array freq.
8. Stop

Program:

```
public class Frequency

{

    public static void main(String[] args) {

      String str = "picture perfect";

      int[] freq = new int[str.length()];

      int i, j;          //Converts given string into character array

      char string[] = str.toCharArray();

           for(i = 0; i <str.length(); i++) {

        freq[i] = 1;
```

```
        for(j = i+1; j <str.length(); j++) {
            if(string[i] == string[j]) {
                freq[i]++;
                                //Set string[j] to 0 to avoid printing visited character
                string[j] = '0';
            }
        }
    }
            //Displays the each character and their corresponding frequency
    System.out.println("Characters and their corresponding frequencies");
    for(i = 0; i <freq.length; i++) {
        if(string[i] != ' ' && string[i] != '0')
            System.out.println(string[i] + "-" + freq[i]);
    }
  }
}
```

Output

```
p-2
i-1
c-2
t-2
```

Exp No. Date:

u-1

r-2

e-3

f-1

Result & Discussion:

Program is executed successfully and output is obtained.

Viva Questions:

1.Explain any 2 OOP Concepts.

2. Give the difference between Entry controlled and exit controlled loops.

3. What is conditional operator?

5. Define Class.

Exp No. Date:

Multiplication of two Matrices

Aim: Write a Java program to multiply two given matrices.

Pseudocode:

```
matrixMultiply(A, B):
Assume dimension of A is (m x n), dimension of B is (p x q)
Begin
  if n is not same as p, then exit
  otherwise define C matrix as (m x q)
  for i in range 0 to m - 1, do
    for j in range 0 to q – 1, do
      for k in range 0 to p, do
        C[i, j] = C[i, j] + (A[i, k] * A[k, j])
      done
    done
  done
End
```

Program:

```
public class MatrixMultiplicationExample{
public static void main(String args[]){
//creating two matrices
```

```
int a[][]={{1,1,1},{2,2,2},{3,3,3}};

int b[][]={{1,1,1},{2,2,2},{3,3,3}};

    //creating another matrix to store the multiplication of two matrices

int c[][]=new int[3][3];  //3 rows and 3 columns

  //multiplying and printing multiplication of 2 matrices

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

c[i][j]=0;

for(int k=0;k<3;k++)

{

c[i][j]+=a[i][k]*b[k][j];

}//end of k loop

System.out.print(c[i][j]+" ");  //printing matrix element

}//end of j loop

System.out.println();//new line

}

}}
```

Output

The first matrix is:

2 4 1

2 3 9

3 1 8

The second matrix is:

1 2 3

3 6 1

2 4 7

Product of the two matrices is:

16 32 17

29 58 72

22 44 66

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.Describe Array Initialization

2. Execution sequence of for loop.

3. Explain array declaration.

4. Differentiate between arrays and structures.

5. What is multidimensional arrays.

Exp No. Date:

Inheritance

Aim: Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'print-Salary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. (Exercise to understand inheritance).

Algorithm:

```
class b {

// implementation of inheritedMethod()

}

class a extends b

{

  inheritedMethod();

}
```

Program:

```
class Employee{

 float salary=40000;

}

class Programmer extends Employee{

 int bonus=10000;

 public static void main(String args[]){
```

```
  Programmer p=new Programmer();

  System.out.println("Programmer salary is:"+p.salary);

  System.out.println("Bonus of Programmer is:"+p.bonus);

}

}
```

Output

Programmer salary is:40000.0

Bonus of programmer is:10000

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.What is inheritance?

2.How inheritance can be specified in Program.

3. Give the significance of ‘extends’

4. Different types of inheritance

5.Explain multiple inheritance in java.

Exp No. Date:

Polymorphism

Aim: Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures. (Exercise to understand polymorphism).

Pseudocode:

```
class Shape{
void draw(){System.out.println("drawing...");}
}
class Rectangle extends Shape{
void draw(){System.out.println("drawing rectangle...");}
}
class Circle extends Shape{
void draw(){System.out.println("drawing circle...");}
}
class Triangle extends Shape{
void draw(){System.out.println("drawing triangle...");}
}
class TestPolymorphism2{
public static void main(String args[]){
Shape s;
s=new Rectangle();
```

```
s.draw();

s=new Circle();

s.draw();

s=new Triangle();

s.draw();

}

}
```

Program:

```
class Shape{

void draw(){System.out.println("drawing...");}

}

class Rectangle extends Shape{

void draw(){System.out.println("drawing rectangle...");}

}

class Circle extends Shape{

void draw(){System.out.println("drawing circle...");}

}

class Triangle extends Shape{

void draw(){System.out.println("drawing triangle...");}

}

class TestPolymorphism2{
```

```
public static void main(String args[]){

Shape s;

s=new Rectangle();

s.draw();

s=new Circle();

s.draw();

s=new Triangle();

s.draw();

}  }
```

Output

```
drawing rectangle...

drawing circle...

drawing triangle...
```

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.Define Polymorphism.

2.Explain types of Polymorphism.

3. super keyword in java.

4. Method overloading in java.

5. this operator in java.

Exp No. Date:

Exception Handling

Aim: Write a Java program that read from a file and write to file by handling all file related exceptions.

Pseudocode:

```
>>> anumber = int(input("Please enter an integer "))
Please enter an integer -23
>>> print(math.sqrt(anumber))
Traceback (most recent call last):
  File "<pyshell#102>", line 1, in <module>
    print(math.sqrt(anumber))
ValueError: math domain error
>>> try:
        print(math.sqrt(anumber))
    except:
        print("Bad Value for square root")
        print("Using absolute value instead")
        print(math.sqrt(abs(anumber)))
Bad Value for square root
Using absolute value instead
4.79583152331
>>>
```

```
>>> if anumber < 0:
 raise RuntimeError("You can't use a negative number")
else:
 print(math.sqrt(anumber))
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
RuntimeError: You can't use a negative number
```

Program:

```
 class Main {
public static void main(String args[]) {
//try block
try{
System.out.println("::Try block::");
int num=67/0;
System.out.println(num);
}
//catch block
catch(ArithmeticException e){
System.out.println("::Catch block::");
System.out.println("ArithmeticException::Number divided by zero");
}
//finally block
finally{
System.out.println("::Finally block::");
}
System.out.println("Rest of the code continues...");
}
```

```
}
```

Throws Programs

```
import java.io.*;
class Main {
// declareing the type of exception
public static void findFile() throws IOException {
// code that may generate IOException
File newFile = new File("test.txt");
FileInputStream stream = new FileInputStream(newFile);
}
public static void main(String[] args) {
try {
findFile();
}
catch (IOException e) {
System.out.println(e);
}
}
}
```

Output

1. statement 1;
2. statement 2;
3. statement 3;
4. statement 4;
5. statement 5;//exception occurs
6. statement 6;
7. statement 7;
8. statement 8;
9. statement 9;
10. statement 10;

Exp No. Date:

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.What is Exception?

2. Significance of throw and catch.

3. How we can define an exception?

4. Give examples for exceptions.

5. Explain method overriding.

Exp No. Date:

String Tokenizer class of java.util

Aim: Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util).

Pseudocode:

```
Scanner sc = new Scanner(System.in);
    System.out.println("Enter integers with one space gap:");
    String s = sc.nextLine();
    StringTokenizer st = new StringTokenizer(s, " ");
    while (st.hasMoreTokens()) {
       String temp = st.nextToken();
       n = Integer.parseInt(temp);
       System.out.println(n);
       sum = sum + n;
```

Program:

```
import java.util.*;
class StringTokenizerDemo {
   public static void main(String args[]) {
      int n;
      int sum = 0;
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter integers with one space gap:");
      String s = sc.nextLine();
```

```
        StringTokenizer st = new StringTokenizer(s, " ");
        while (st.hasMoreTokens()) {
            String temp = st.nextToken();
            n = Integer.parseInt(temp);
            System.out.println(n);
            sum = sum + n;
        }
        System.out.println("sum of the integers is: " + sum);
        sc.close();
    }
}
```

Output

Enter integers with one space gap:

10 20 30 40 50

10

20

30

40

50

sum of the integers is: 150

Exp No. Date:

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1. Define package in java.

2. How we can create new package in java?

3. Advantages of defining packages in java.

4. What is the package util in java?

5.What is the class string Tokenizer?

Exp No. Date:

Thread Synchronization

Aim: Write a Java program that shows thread synchronization.

pseudocode:

```
class ThreadDemo extends Thread {
   private Thread t;
   private String threadName;
   PrintDemo  PD;

   ThreadDemo( String name,  PrintDemo pd) {
      threadName = name;
      PD = pd;
   }

   public void run() {
      PD.printCount();
      System.out.println("Thread " +  threadName + " exiting.");
   }
   public void start () {
      System.out.println("Starting " +  threadName );
      if (t == null) {
         t = new Thread (this, threadName);
```

```
        t.start ();
      }
   }
```

Program:

```
class PrintDemo {
   public void printCount() {
      try {
         for(int i = 5; i > 0; i--) {
            System.out.println("Counter   ---   "  + i );
         }
      } catch (Exception e) {
         System.out.println("Thread  interrupted.");
      } }}
class ThreadDemo extends Thread {
   private Thread t;
   private String threadName;
   PrintDemo  PD;
   ThreadDemo( String name,  PrintDemo pd) {
      threadName = name;
      PD = pd;
   }
```

```
    public void run() {
    PD.printCount();
    System.out.println("Thread " +  threadName + " exiting.");
  }

  public void start () {
    System.out.println("Starting " +  threadName );
    if (t == null) {
      t = new Thread (this, threadName);
      t.start ();
    } }}
public class TestThread {
  public static void main(String args[]) {
    PrintDemo PD = new PrintDemo();
    ThreadDemo T1 = new ThreadDemo( "Thread - 1 ", PD );
    ThreadDemo T2 = new ThreadDemo( "Thread - 2 ", PD );
    T1.start();
    T2.start();
    // wait for threads to end
      try {
      T1.join();
```

```
        T2.join();
      } catch ( Exception e) {
        System.out.println("Interrupted");
      }
   }
}
```

Output

```
Starting Thread - 1
Starting Thread - 2
Counter   ---   5
Counter   ---   4
Counter   ---   3
Counter   ---   5
Counter   ---   2
Counter   ---   1
Counter   ---   4
Thread Thread - 1  exiting.
Counter   ---   3
Counter   ---   2
Counter   ---   1
Thread Thread - 2  exiting.
```

Exp No. Date:

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.Explain thread synchronization.

2.Define thread.

3.Describe the concept of multiple threads.

4. What is monitor in java?

5. Define the term “synchronized” in java.

Exp No. Date:

usage of try, catch, throws and finally

Aim: Write a Java program that shows the usage of try, catch, throws and finally.

pseudocode:

```
try {
   System.out.println("Access element three :" + a[3]);
} catch (ArrayIndexOutOfBoundsException e) {
   System.out.println("Exception thrown :" + e);
} finally {
   a[0] = 6;
   System.out.println("First element value: " + a[0]);
   System.out.println("The finally statement is executed");
}
```

Program:

```
public class ExcepTest {
  public static void main(String args[]) {
    int a[] = new int[2];
    try {
      System.out.println("Access element three :" + a[3]);
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("Exception thrown :" + e);
    } finally {
```

```
        a[0] = 6;
        System.out.println("First element value: " + a[0]);
        System.out.println("The finally statement is executed");
      }
   }
}
```

Output

Exception thrown :java.lang.ArrayIndexOutOfBoundsException: 3

First element value: 6

The finally statement is executed

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1. When to use throws throw VS try-catch in Java?

2. Can we use throws, try and catch in a single method?

3. What happens when a catch block throws an exception?

4. What is try-catch-finally in Java?

5. Can finally block have try-catch?

Exp No. Date:

Simple Calculator

Aim: Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing.

pseudocode:

inputData = Read_Input() result = Perform_Calculations(inputData) Display_Results(result)

if operator = "+" result = firstNumber + secondNumber
else if operator = "-" result = firstNumber - secondNumber
else if operator = "*" result = firstNumber * secondNumber
else if operator = "/" result = firstNumber / secondNumber
Display_Result(result)

Program:

```
//Program for implementing a Simple Calculator

import java.awt.*;

import java.awt.event.*;

import java.applet.*;

/*<applet  code="Calculator1" width=300 height=300></applet>*/

public class Calculator1 extends Applet implements ActionListener

{

        TextField t;

        Button b[]=new Button[15];

        Button b1[]=new Button[6];
```

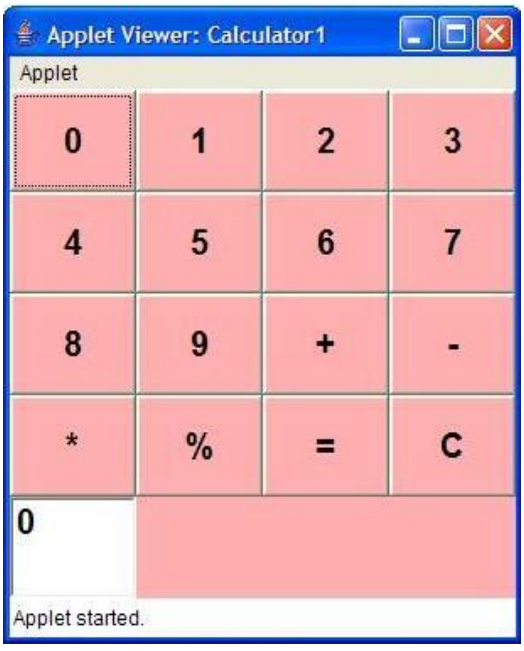
```
String op2[]={"+","-","*","%","=","C"};
String str1="";
int p=0,q=0;
String oper;
public void init()
{
    setLayout(new GridLayout(5,4));
    t=new TextField(20);
    setBackground(Color.pink);
    setFont(new Font("Arial",Font.BOLD,20));
    int k=0;
    t.setEditable(false);
    t.setBackground(Color.white);
    t.setText("0");
    for(int i=0;i<10;i++)
    {
        b[i]=new Button(""+k);
        add(b[i]);
        k++;
        b[i].setBackground(Color.pink);
        b[i].addActionListener(this);
```

```
        }

        for(int i=0;i<6;i++)
        {
            b1[i]=new Button(""+op2[i]);
            add(b1[i]);
            b1[i].setBackground(Color.pink);
            b1[i].addActionListener(this);
        }
        add(t);
    }
    public void actionPerformed(ActionEvent ae)
    {

        String  str=ae.getActionCommand();
                    if(str.equals("+")){      p=Integer.parseInt(t.getText());
                        oper=str;
                        t.setText(str1="");
                    }
        else if(str.equals("-")){  p=Integer.parseInt(t.getText());
                        oper=str;
```

```
                t.setText(str1="");
              }
else if(str.equals("*")){  p=Integer.parseInt(t.getText());
                oper=str;
               t.setText(str1="");
              }
else if(str.equals("%")){  p=Integer.parseInt(t.getText());
                 oper=str;
                t.setText(str1="");
            }
else if(str.equals("=")) { str1="";
                if(oper.equals("+"))  {
                q=Integer.parseInt(t.getText());
                t.setText(String.valueOf((p+q)));}

                else if(oper.equals("-")) {
                q=Integer.parseInt(t.getText());
                t.setText(String.valueOf((p-q))); }
                else  if(oper.equals("*")){
                q=Integer.parseInt(t.getText());
                t.setText(String.valueOf((p*q))); }
```

```
                                        else if(oper.equals("%")){
                                        q=Integer.parseInt(t.getText());
                                        t.setText(String.valueOf((p%q))); }
                                    }
            else if(str.equals("C")){  p=0;q=0;
                                        t.setText("");
                                        str1="";
                                        t.setText("0");
                                    }
        else{  t.setText(str1.concat(str));
                                    str1=t.getText();
                                } }                    }
```

Output



Exp No. Date:

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1.Explain awt package in java.

2.What is swing in java?

3.Describe applets in java.

4. What do you meant by event handling in java?

5.What is actionlistener in java?

Exp No. Date:

Event Handling

Aim: Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts.

pseudocode:

```
while (true) {                          // The event loop.
    // Get the next event from the event queue.
    Event e = get_next_event();

    // Process the events by calling appropriate event handlers.
    if (e.eventType == QUIT) {
       exit();                          // Terminate the program.
    }
    else if (e.eventType == BUTTON_PUSHED) {
       if (e.eventSource == PRINT_BUTTON)
          print(e);                     // Print out the current page.
       else {
          ...
       }
    }
    else {
       ...
    }
  }
```

Program:

```
import java.awt.event.*;

import java.applet.*;

import java.awt.*;
```

Exp No. Date:

```
/*
  <applet code="TrLight" width=250 height=350>
  </applet>
*/
public class TrLight extends Applet implements ItemListener {
  String msg = "";
  Checkbox red,green,yellow;
  CheckboxGroup cbg;
  public void init() {
    cbg = new CheckboxGroup();
    red = new Checkbox("Red", cbg, false);
    green = new Checkbox("Green", cbg, false);
    yellow = new Checkbox("Yellow", cbg, false);
    add(red);
    add(yellow);
    add(green);
    red.addItemListener(this);
    yellow.addItemListener(this);
    green.addItemListener(this);
  }
```

```
public void itemStateChanged(ItemEvent ie) {
  repaint();
}
public void paint(Graphics g) {
   Color color;
   color=Color.BLACK;
   g.setColor(color);
   g.drawOval(50, 50, 52, 52);
   g.drawOval(50, 103, 52, 52);
   g.drawOval(50, 156, 52, 52);
   String col = cbg.getSelectedCheckbox().getLabel();
   System.out.println(col);
   if(col.equalsIgnoreCase("Green"))
  {
   color= Color.GREEN;
   g.setColor(color);
   g.fillOval(50, 156, 52, 52);
  }
   if(col.equalsIgnoreCase("Red"))
  {
```

```
    color=Color.RED;
    g.setColor(color);
    g.fillOval(51, 51, 51, 51);
   }
    if(col.equalsIgnoreCase("Yellow"))
   {
    color=Color.YELLOW;
    g.setColor(color);
    g.fillOval(50, 103, 51, 51);
   }
  }
}
```

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1 Explain JDK, JRE and JVM?

2. Explain public static void main(String args[]) in Java

3. Why Java is platform independent?

4. Why Java is not 100% Object-oriented?

5. What are wrapper classes in Java?

Exp No. Date:

Doubly Linked List

Aim: Write a Java program to Create a doubly linked list of elements.

Algorithm:

Define a Node class which represents a node in the list. It will have three properties: data, previous which will point to the previous node and next which will point to the next node.

Define another class for creating a doubly linked list, and it has two nodes: head and tail. Initially, head and tail will point to null.

addNode() will add node to the list:

It first checks whether the head is null, then it will insert the node as the head.

Both head and tail will point to a newly added node.

Head's previous pointer will point to null and tail's next pointer will point to null.

If the head is not null, the new node will be inserted at the end of the list such that new node's previous pointer will point to tail.

The new node will become the new tail. Tail's next pointer will point to null.

a. display() will show all the nodes present in the list.

Define a new node 'current' that will point to the head.

Print current.data till current points to null.

Current will point to the next node in the list in each iteration.

Program:

```
public class DoublyLinkedList {

    //Represent a node of the doubly linked list

    class Node{

      int data;
```

```
    Node previous;
    Node next;
      public Node(int data) {
        this.data = data;
      }
  }
   //Represent the head and tail of the doubly linked list
  Node head, tail = null;
   //addNode() will add a node to the list
  public void addNode(int data) {
    //Create a new node
    Node newNode = new Node(data);
     //If list is empty
    if(head == null) {
      //Both head and tail will point to newNode
      head = tail = newNode;
      //head's previous will point to null
      head.previous = null;
      //tail's next will point to null, as it is the last node of the list
      tail.next = null;
    }
```

```
        else {
            //newNode will be added after tail such that tail's next will point to newNode
            tail.next = newNode;
            //newNode's previous will point to tail
            newNode.previous = tail;
            //newNode will become new tail
            tail = newNode;
            //As it is last node, tail's next will point to null
            tail.next = null;
        }
    }
     //display() will print out the nodes of the list
    public void display() {
        //Node current will point to head
        Node current = head;
        if(head == null) {
            System.out.println("List is empty");
            return;
        }
        System.out.println("Nodes of doubly linked list: ");
        while(current != null) {
```

```
        //Prints each node by incrementing the pointer.
          System.out.print(current.data + " ");
        current = current.next;
    }   }
  public static void main(String[] args) {
    DoublyLinkedList dList = new DoublyLinkedList();
    //Add nodes to the list
    dList.addNode(1);
    dList.addNode(2);
    dList.addNode(3);
    dList.addNode(4);
    dList.addNode(5);
      //Displays the nodes present in the list
    dList.display();
} }
```

Output

Nodes of doubly linked list

1 2 3 4 5

Result & Discussion

Program is executed successfully and output is obtained.

Exp No. Date:

Viva Questions:

1. What is abstraction in Java?

2. What do you mean by an interface in Java?

3. What is the difference between abstract classes and interfaces?

4. What is inheritance in Java?

5. What are the different types of inheritance in Java?

Exp No. Date:

Quick sort

Aim: Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order.

Pseudocode:

```
/* low  --> Starting index,  high  --> Ending index */

quickSort(arr[], low, high)

{

   if (low < high)

   {

      /* pi is partitioning index, arr[p] is now

         at right place */

      pi = partition(arr, low, high);



      quickSort(arr, low, pi - 1);  // Before pi

      quickSort(arr, pi + 1, high); // After pi

   }

}
```

Program:

```
import java.util.Arrays;

class Quicksort {

  // method to find the partition position

  static int partition(int array[], int low, int high) {
```

```
    // choose the rightmost element as pivot
  int pivot = array[high];
    // pointer for greater element
  int i = (low - 1);
  // traverse through all elements
  // compare each element with pivot
  for (int j = low; j < high; j++) {
   if (array[j] <= pivot) {
    // if element smaller than pivot is found
    // swap it with the greater element pointed by i
    i++;
    // swapping element at i with element at j
    int temp = array[i];
    array[i] = array[j];
    array[j] = temp;
   }
  }
  // swap the pivot element with the greater element specified by i
  int temp = array[i + 1];
  array[i + 1] = array[high];
  array[high] = temp;
```

```
    // return the position from where partition is done
    return (i + 1);
  }
  static void quickSort(int array[], int low, int high) {
    if (low < high) {
      // find pivot element such that
      // elements smaller than pivot are on the left
      // elements greater than pivot are on the right
      int pi = partition(array, low, high);
            // recursive call on the left of pivot
      quickSort(array, low, pi - 1);
      // recursive call on the right of pivot
      quickSort(array, pi + 1, high);
    } } }
// Main class
class Main {
  public static void main(String args[]) {
    int[] data = { 8, 7, 2, 1, 0, 9, 6 };
    System.out.println("Unsorted Array");
    System.out.println(Arrays.toString(data));
```

```
    int size = data.length;

    // call quicksort() on array data

    Quicksort.quickSort(data, 0, size - 1);

    System.out.println("Sorted Array in Ascending Order ");

    System.out.println(Arrays.toString(data));

  }

}
```

Output

Unsorted Array

[8, 7, 2, 1, 0, 9, 6]

Sorted Array in Ascending Order

[0, 1, 2, 6, 7, 8, 9]

Result & Discussion

Program is executed successfully and output is obtained.

Viva Questions:

1. What are constructors in Java?
2. What is singleton class in Java and how can we make a class singleton?
3. What is the difference between Array list and vector in Java?
4. What is the difference between equals() and == in Java?
5. What are the differences between Heap and Stack Memory in Java?